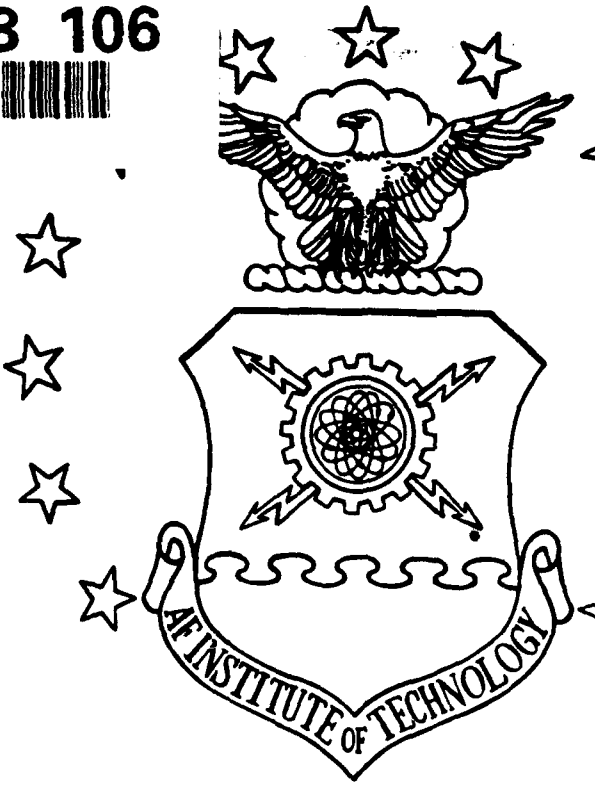

AFIT/EN/TR93-10

Air Force Institute of Technology

Computing with Bayesian Multi-Networks

Eugene Santos Jr.

November 16, 1993

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

UNCLASSIFIED

AFIT/EN/TR93-10

Air Force Institute of Technology

Computing with Bayesian Multi-Networks

Eugene Santos Jr.

November 16, 1993

Approved for public release; distribution unlimited

Computing with Bayesian Multi-Networks

Eugene Santos Jr.
Department of Electrical and Computer Engineering
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433-7765
esantos@afit.af.mil

November 16, 1993

KEYWORDS: probabilistic reasoning, constraint satisfaction, linear programming, temporal reasoning, abductive explanation

Abstract

Existing probabilistic approaches to automated reasoning impose severe restrictions on its knowledge representation scheme. Mainly, this is to ensure that there exists an effective inferencing algorithm. Unfortunately, this makes the application of these approaches to general domains quite difficult.

In this paper, we present a new model called Bayesian multi-networks which uses a rule-based organization of knowledge quite natural for human experts modeling various domains. Furthermore, strong probabilistic semantics help quantify the knowledge. Combined with the rich structure of rule-based approaches, a general inference engine for Bayesian multi-networks is developed.

1 Introduction

The success of automated reasoning will clearly depend on its applicability to a wide variety of problem domains. It must have a flexible knowledge representation scheme as well as provide effective and efficient inference mechanisms. Unfortunately, knowledge representation and inference algorithms always seem to be at odds with one another. This is particularly true for probabilistic approaches which are often trading off flexibility for efficiency and even effective-

ness. The major problem facing them, regardless of whether they are being used for story understanding, robotic planning, diagnosis, expert systems, etc., is the difficulty of modeling the desired problem domain. More often than not, severe restrictions are placed on representation just to ensure that the inference algorithm employed at least finishes in less than combinatorial time once in awhile. Of course, the net result of these restrictions is the increased frustration of trying to build such knowledge bases beyond the simplest of domains. The biggest problem rests in re-interpreting (human) expert knowledge to fit the restricted representation and quantifying all the required information in some reasonable fashion.

One of the most popular probabilistic approaches is *Bayesian networks* [7]. It offers a high level of readability by providing a clear graphical representation while allowing for efficient computations on certain subclasses of networks [7, 19, 9, 12, 14, 5, 1]. Although it has done much for the probabilistic paradigm, it still falls far short of the desired level of knowledge representation.

One such short fall has been identified in [4, 3] called *assymetric independence*. It asserts that random variables (abbrev. r.v.s) are dependent for some but not necessarily all of their values. Hence, a fair amount of redundancy occurs in the conditional tables of the network. By identifying them, it is hoped that the number of conditional probabilities can be reduced thus decreasing the computational time required.

In a separate research effort, [16, 15] has also identified a similar problem. Although this work was approached from the standpoint of solving the *over-specification* problem when performing belief revision on Bayesian networks, it nevertheless identifies that there exists situations where a particular instantiation to a r.v. will render the remaining r.v. in the conditional "irrelevant". In other words, there is little or no change in the final conditional probability when the instantiations to the irrelevant r.v. are altered.

Although these earlier results have served to increase the expressiveness of Bayesian networks, they are still fairly limited. If we look carefully at both approaches, we find that they are still working entirely within the realm of Bayesian network restrictions. Simply put, these methods take a Bayesian network and create a set of smaller networks which are actually subnetworks of the original. Thus, they must still for the most part obey the restrictions.

To reiterate, the most important aspects to having a successful probabilistic model are:

- The ease in which knowledge can be encoded and decoded by a human expert.
- The quantifiability of the knowledge encoded.
- The existence of an effective and efficient inference engine.

Probably the best knowledge representation scheme satisfying the above conditions would be the rule-based approaches. [18] points out in their milestone work on MYCIN that the human experts have found that working with rule-based information is much more natural than working with any other schemes.

Furthermore, these rules are much more easily quantified and provide a rich structure for inferencing.

In this paper, we present a new model called *Bayesian multi-networks*.[1] This model permits a general rule-based organizational scheme while providing a strong probabilistic semantics for the quantified rules. Furthermore, a general inference algorithm can be found for computing with these multi-networks.

We begin in Section 2 by describing our Bayesian multi-networks. Section 3 describes our inference algorithm. Finally, Section 4 discusses consistency considerations when constructing a Bayesian multi-network.

2 Bayesian Multi-networks

Bayesian networks organize knowledge in terms of probabilistic conditional dependencies. The world can be modeled by a collection of events and the relationships between them. In particular, both causal and logical relationships are represented in terms of conditional dependencies. For example, given two events A and B where event A represents "It is raining." and B represents "The sidewalk is wet.", the relationship that A *causes* B is denoted by $P(B = \text{true} \,|A = \text{true})$. We say that B is *conditionally dependent* on A.

We can redescribe this more clearly in terms of graphs. A Bayesian network is a directed acyclic graph whose nodes are r.v.s and whose arcs between the nodes represent direct conditional dependencies between the r.v.s. Given a node A with parents $B_1, \ldots, B_n$, we say that A is conditionally dependent on the set of r.v.s $B_1, \ldots, B_n$. Furthermore, Bayesian networks assert that if $B_1, \ldots, B_n$ are all the parents of A, then

$$P(A = a|B_1 = b_1, \ldots, B_n = b_n) = \\ P(A = a|B_1 = b_1, \ldots, B_n = b_n, C_1 = c_1, \ldots, C_m = c_m) \tag{1}$$

for any collection of r.v.s $\{C_1, \ldots, C_m\}$ such that

$$\{C_1, \ldots, C_m\} \cap \{A, B_1, \ldots, B_n\} = \Phi$$

where Φ denotes the empty set and for any instantiation to all the r.v.s involved.[2] We will see the significance of this assertion later in this section.

The r.v.s are used to represent events in the world. An *instantiation* of a r.v. to some value reflects the status of the event being modeled. For example, instantiating A to true above implies that it is raining. Hence, we can represent the various states of the world (scenarios) through different instantiations to the r.v.s. Furthermore, we can properly attach a probability to a given scenario representing the likelihood of the scenario. The question now arises as to how we

[1] Different from the multinets in [3].

[2] Note that this isn't quite true but suffices for our discussion. For more information, see the notion of d-separation in [7].

go about computing this probability. This is where the above assertion comes into play.

Consider the scenario $\{C_1 = c_1, \ldots, C_n = c_n\}$ which instantiates all the r.v.s in the network. We must somehow compute its joint probability. Let's assume that $\{C_1, C_2, \ldots, C_n\}$ already are a topological ordering of the r.v.s consistent with the associated directed acyclic graph for the Bayesian network. We first apply Bayes' theorem and get

$$\begin{aligned} P(C_1 = c_1, \ldots, C_n = c_n) = \\ P(C_1 = c_1 | C_2 = c_2, \ldots, C_n = c_n) \\ P(C_2 = c_2 | C_3 = c_3, \ldots, C_n = c_n) \\ \ldots \\ P(C_n = c_n). \end{aligned} \tag{2}$$

From the assertion above on conditional dependencies, we can then substitute the conditional probabilities with smaller ones from (1) reducing the problem to one of simply multiplying the appropriate conditional probabilities together from our network.

One of the computations performed on Bayesian networks is *belief revision*. It is also called the search for the *most-probable explanation* and is characterized as follows: Given a set of instantiations e called *evidence* on some subset of the r.v.s, find an instantiation w^* to all the r.v.s such that

$$P(w^*|e) = \max_w P(w|e)$$

Furthermore, belief revision is a model for abductive reasoning[2]. We call the instantiation-sets w *complete instantiation sets* or *complete scenarios*. They are also called *explanations* for e.

NOTATION. Given a r.v. A, $R(A)$ represents the set of all possible instantiations to A.

NOTATION. Given a conditional probability $P(A = a|B_1 = b_1, \ldots, B_n = b_n)$, we define

$$\mathsf{head}(P(\ldots)) = \{A = a\}$$

$$\mathsf{tail}(P(\ldots)) = \{B_1 = b_1, \ldots, B_n = b_n\}$$

In essence, we can simply describe a Bayesian network as a database of conditional probabilities which satisfy certain conditions.

1. For any $P(A = a|B_1 = b_1, \ldots, B_n = b_n)$ in the database, the following condition holds:

$$\begin{aligned} P(A = a|B_1 = b_1, \ldots, B_n = b_n) = \\ P(A = a|B_1 = b_1, \ldots, B_n = b_n, C_1 = c_1, \ldots, C_m = c_m) \end{aligned} \tag{3}$$

for any[3] $\{C_1 = c_1, \ldots, C_m = c_m\}$ such that

$$\{C_1, \ldots, C_m\} \cap \{A, B_1, \ldots, B_n\} = \Phi.$$

2. For each r.v. A, there exists a unique collection of r.v.s $\{B_1, \ldots, B_n\}$ such that all conditional probabilities in the database are of the form:

$$P(A = a | B_1 = b_1, \ldots, B_n = b_n).$$

3. For each r.v. A conditioned on $\{B_1, \ldots, B_n\}$, for each $a \in R(A), b_1 \in R(B_1), \ldots, b_n \in R(B_n)$,

$$P(A = a | B_1 = b_1, \ldots, B_n = b_n) \tag{4}$$

is in the database.

4. We can find a topological ordering on the r.v.s based on the conditional probabilities in the database.

The first condition is simply our conditional dependency assertion. The second condition reflects the requirement of Bayesian networks that they be built around r.v.s. In particular, independencies are between r.v.s and not between the different instantiations of the r.v.s. The third require that the tables be complete while the fourth requires our network to be acyclic in nature.

A major representational restriction of Bayesian networks arise from the second and fourth conditions as we mentioned earlier in our introduction. Our approach is to eliminate this restriction in order to accommodate a more powerful representational scheme. In particular, we wish to be able to represent the following information: Given r.v.s A, B, C, D, $A = a_1$ is only conditionally dependent on $B = b$ and $C = c_1$ whereas $C = c_2$ is only conditionally dependent on $A = a_2$ and $D = d$ where $a_1 \neq a_2$ and $c_1 \neq c_2$. These conditions correspond to the rules

$$\{B = b\} \wedge \{C = c_1\} \longrightarrow \{A = a_1\}$$
$$\{A = a_2\} \wedge \{D = d\} \longrightarrow \{C = c_2\}$$

Clearly, this organization violates both the second and fourth condition of Bayesian networks.

We now formalize our Bayesian multi-networks.

DEFINITION 2.1. *A* Bayesian multi-network $\mathcal{M}$ *is an ordered pair* $(\mathcal{V}, \mathcal{P})$ *where* $\mathcal{V}$ *is a finite set of r.v.s and* $\mathcal{P}$ *is a finite collection of conditional probabilities on* $\mathcal{V}$ *such that for all probabilities* $P(A = a | B_1 = b_1, \ldots, B_n = b_n) \in \mathcal{P}$,

$$P(A = a | B_1 = b_1, \ldots, B_n = b_n, C_1 = c_1, \ldots, C_m = c_m) = P(A = a | B_1 = b_1, \ldots, B_n = b_n)$$

[3] Again, see the notion of d-separation.

for any instantiations to $\{C_1, \ldots, C_m\} \subseteq \mathcal{V}$ *and*

$$\{C_1, \ldots, C_m\} \cap \{A, B_1, \ldots, B_n\} = \Phi$$

We have defined our multi-network in terms of a database of conditional probabilities. Obviously, given the conditional dependency restriction, we can still calculate joints probabilities similar to what we did before for Bayesian networks. We simply pick an appropriate collection of conditional probabilities from the database such that it satisfies (2). Intuitively, we can view

$$P(A = a | B_1 = b_1, \ldots, B_n = b_n)$$

in the database as the information that

$$\{A = a\} \text{ is supported by } \{B_1 = b_1, \ldots, B_n = b_n\}.$$

Hence, the collection of conditional probabilities used in calculating a joint probability represents a causal inference chain. In a Bayesian network, there is exactly one such chain which is dictated by the directed acyclic ordering of all the r.v.s. For our multi-networks, the ordering is asserted at a finer level than just between r.v.s. It is ordered with respect to particular instantiations of each r.v.s. In essence, this allows us to have multiple causal chains. Hence, we have a multi-network.

Furthermore, like Bayesian networks, multi-networks also have a graphical form. Instead of nodes representing r.v.s, they will represent individual r.v. instantiations such as $\{A = a\}$. More specifically, we have two types of nodes in the multi-network graph. The first type represents r.v. instantiations whereas the second type is used to explicitly represent the one or more different ways a particular r.v. instantiation can be supported. The parents of an *instantiation node* are *support nodes*. Each support node corresponds to a conditional probability in the database whose head is the instantiation node and whose tail are the parents of the support node. See Figure 2.1.

As we can easily see, Bayesian networks are simply a special case of Bayesian multi-networks. We now briefly mention a special class of multi-networks whose properties will be particularly useful for our computation techniques in the next sections.

DEFINITION 2.2. *A Bayesian multi-network* $\mathcal{M} = (\mathcal{V}, \mathcal{P})$ *is said to be* anti-cyclic *if and only if there exists a topological ordering on all the individual r.v. instantiations.*

According to this definition, it follows straightforwardly that Bayesian networks are also anti-cyclic multi-networks.

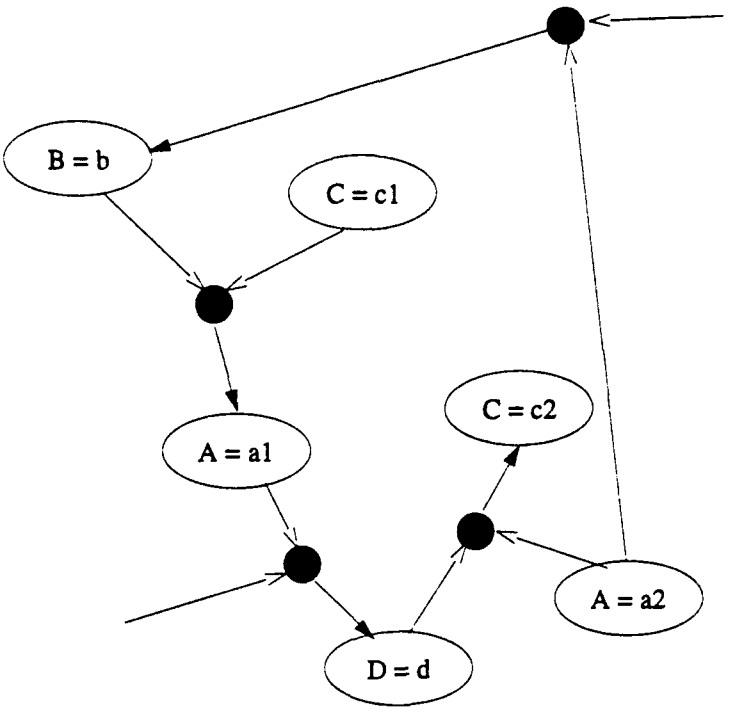

FIG. 2.1. Building a graph for a Bayesian multi-network.

3 Computing Explanations

We now consider belief revision computations on our multi-networks similar to those for normal Bayesian networks. In Section 2, given a set of instantiations e called evidence, we search for a complete instantiation set w^* which maximizes

$$\max_w P(w|e).$$

However, determining w^* which is the most-probable explanation for e on Bayesian networks requires that the conditional probability tables be complete. (See condition 3 for Bayesian networks in the previous section.) Existing methods such as message-passing schemes [7, 19] rely on this property to "localize" the computations in the network.

Still, we can readily define a best explanation for an incomplete system (either Bayesian networks or multi-networks) as follows: We say that a complete instantiation set w is *computable* if there exists a collection of conditional probabilities from our database which can be multiplied to compute the joint probability from (2). This works because of the conditional dependency assertion we placed on the conditional probabilities in our database. We can then say that the best explanation is really the *most probable computable explanation* (abbrev., MPCE).

Even if our multi-networks are probabilistically complete, almost all existing methods for belief revision on Bayesian networks are not extendable to our new model. We have already mentioned the locality restriction required by message-passing schemes above. Another more serious restriction is acyclicity of the information [7, 19, 17]. Our multi-networks can contain directed cycles.

The one method which can be extended from Bayesian networks to our new multi-networks is Santos' linear programming method [9, 1, 10, 11]. Basically, the search for the MPCE is transformed into an integer linear programming problem.

The transformation involves mapping r.v. instantiations into some multi-dimensional space which we will denote by $\Re^n$. A subspace of $\Re^n$ will represent "valid" instantiations where valid includes things like being consistent to the given evidence e, each r.v. has at most one instantiation, etc. In particular, we are interested in transforming it into a *polyhedral convex set.*[4] Such a set can be described by a collection of linear inequalities. As it turns out, these inequalities will intuitively correspond to the restrictions/constraints required in making valid instantiations of the r.v.s. Finally, we would like to define a *linear energy function* such that by minimizing it over the convex set, the resulting answer will be the best explanation after we make the appropriate inverse mapping. Thus, we would have the makings of a linear constraint satisfaction problem.

[4] "Polyhedral" refers to the fact that the boundaries of the subspace are composed of hyperplanes.

For the time being, let's consider the class of anti-cyclic multi-networks. Let A be a r.v. and a be a possible instantiations for A. If A is instantiated to a, that is, $A = a$, then we would like to set a real variable $\mathsf{A_a}$ to the value 1. If $A \neq a$, then $\mathsf{A_a} = 0$. This holds for every possible instantiation of A for every r.v. A. Next, each r.v. must have exactly one instantiation. This can be achieved with the linear constraint

$$\sum_{a \in R(A)} \mathsf{A_a} = 1 \tag{5}$$

where $R(A)$ is the set of all possible instantiations for A.

Now, we must tie the random variable instantiations to the conditional probabilities in the database. For each $p \in \mathcal{P}$, associate a real variable q_p called a *conditional variable*. When q_p is 1, this implies that the p is being used in the joint probability computation. Otherwise, $q_p = 0$ implies it is not being used.

Clearly, if r.v. A is not instantiated to a, then any conditional probability with $A = a$ in either it's head or tail should not be used.

$$K\,\mathsf{A_a} \geq \sum_{p \in Q(\{A=a\})} q_p \tag{6}$$

where K is some arbitrarily large constant and $Q(\{A = a\})$ is the collection of all conditional probabilities in $\mathcal{P}$ which contain $A = a$.

Finally, since we know that each r.v. A must be instantiated to exactly one element, this implies that exactly one of the conditional probabilities in $\mathcal{P}$ whose head has A must be used.

$$\sum_{p \in H(A)} q_p = 1 \tag{7}$$

where $H(A)$ is the collection of all conditional probabilities in $\mathcal{P}$ whose head contains A.

Taking all these constraints together, we can prove the following theorems:

THEOREM 3.1. *Any 0-1 assignment to the real variables which satisfy the above constraints correspond to a valid complete instantiation of the r.v.s in a anti-cyclic multi-network. Furthermore, the corresponding complete instantiations are computable.*

THEOREM 3.2. *The converse of Theorem 3.1 is also true.*

Evidence is incorporated into the constraints by clamping the associated real variables to 1. For example,

$$\{A = a\} \in e \Longrightarrow \mathsf{A_a} = 1.$$

To complete our transformation, we need to define our objective function we wish to optimize. Obviously, we can directly compute the joint probability

of a complete instantiation by multiplying the associated probabilities of the conditional variables which have been assigned a value of 1. To recast it as a linear function for our integer linear program, we have

$$\sum_{p \in \mathcal{P}} (-\log p) q_p. \tag{8}$$

THEOREM 3.3. *A 0-1 assignment which satisfies the above constraints and minimizes the objective function is a* MPCE *for the original anti-cyclic multi-network.*

Hence, we have transformed our problem for anti-cyclic multi-networks into integer linear programming.

We began by looking at anti-cyclic multi-networks instead of the more general case since the resulting constraint system is much quicker to comprehend. When we have non-anti-cyclic multi-networks, we can encounter the following situation: Given r.v.s A, B and C, assume $\mathcal{P}$ only contains

$$P(A = a|B = b)$$
$$P(B = b|C = c)$$
$$P(C = c|A = a)$$

Clearly, $\mathcal{P}$ is incomplete. However, if we attempt to compute

$$P(A = a, B = b, C = c)$$

using our above method, we would simply multiply the three probabilities together. This is incorrect since

$$P(A = a, B = b, C = c)$$

is not computable. A anti-cyclic multi-network forbids this type of database. Unfortunately, it also precludes databases of the form

$$P(A = a|B = b, D = d_1)$$
$$P(B = b|C = c, D = d_1)$$
$$P(C = c|A = a, D = d_2)$$
$$P(C = c|D = d_1)$$
$$P(A = a|D = d_2)$$
$$P(B = b|D = d_2)$$

where

$$P(A = a, B = b, C = c, D = d_1)$$

and

$$P(A = a, B = b, C = c, D = d_2)$$

are computable.

As it turns out, we can solve this with additional constraints to guarantee a topological ordering among the instantiations. We associate a topological value variable t_{A_a} to each $\{A = a\}$. Basically, this variable will contain a number reflecting a topological ordering. For each pair of instantiations $\{A = a\}$ and $\{B = b\}$, add the following constraints:

- $$Km_{A_a,B_b} \geq \sum_{p \in M(\{A=a\},\{B=b\})} q_p \tag{9}$$

  where $M(\{A = a\}, \{B = b\})$ are all condition probabilities in $\mathcal{P}$ with head $\{A = a\}$ and whose tail contains $\{B = b\}$ and K is an arbitrarily large positive number. m_{A_a,B_b} will function as a flag indicating that $\{A = a\}$ topologically dominates $\{B = b\}$ since one of the probabilities in $M(\{A = a\}, \{B = b\})$ has been used.

- $$K(1 - m_{A_a,B_b}) + t_{A_a} \geq t_{B_b} + 1 \tag{10}$$

Note that our topological variables need not be integer values. Since both topological dominance and "$\geq$" are transitive, these equations will guarantee that any solution will have a topological ordering.

A topological ordering actually corresponds to a proof path. An explanation for e is a proof for e. This is the heart of abductive reasoning [6, 2, 8]. When we have cyclicity, we often end up with circular proofs which are invalid [13]. However, requiring a topological ordering will eliminate these circular cases.

4 Database Consistency Checking

Building a Bayesian multi-network is relatively straightforward since any rule-based construction approach is sufficient and our computational method will work on the entire class. About the only thing to take note of while constructing the probabilistic database is the issue of consistency between the probabilities. In particular, we wish to avoid the possibility of having two different collections of conditional probabilities that when multiplied together result in different values for a single joint distribution. For example, consider the simple scenario where we have the r.v.s A, B and C and we have the following database:

$$P(A = a|B = b)$$
$$P(B = b|C = c)$$
$$P(C = c)$$
$$P(B = b|A = a)$$
$$P(A = a|C = c)$$

As we can easily see, we can compute $P(A = a, B = b, C = c)$ in two different ways.

We now provide here a relatively straight-forward algorithm for checking database consistency. It exploits our conditional dependency assertion: For any two probabilities p_1, p_2 in $\mathcal{P}$, if $\mathsf{head}(p_1) = \mathsf{head}(p_2)$ and there does not exist a r.v. C in $\mathsf{tail}(p_1)$ and $\mathsf{tail}(p_2)$ such that $\{C = c_1\} \in \mathsf{tail}(p_1)$ and $\{C = c_2\} \in \mathsf{tail}(p_2)$ where $c_1 \neq c_2$, then the probabilities for p_1 and p_2 must be the same.

Clearly, this is a very crude algorithm and many optimizations can be made. However, we can prove that it is sufficient to guarantee that our database is probabilistically consistent and can be computed in $O(|\mathcal{P}|^2|\mathcal{V}|)$ time.

5 Conclusion

Traditional probabilistic approaches to automated reasoning have suffered from representational inadequacies in order to provide effective inference mechanisms. In particular, the restrictions imposed have rendered it quite difficult to build such knowledge bases let alone quantify them.

In this paper, we have developed a new framework called Bayesian multi-networks. This approach provides an extremely natural method for organizing our knowledge through a rule-based scheme. Little or no restrictions are imposed that may hinder the construction of such a network for use in almost any domain. Furthermore, it provides a graphical representation which permits an easier visualization of the information to the human expert.

Strong probabilistic semantics aid in the quantification of information. Combined with the rich structure inherent in a rule-based approach, a general inference engine was developed using linear programming techniques.

Bayesian networks is one of the most popular approaches for automated reasoning. Our new Bayesian multi-networks formulation subsumes Bayesian networks. Because of this, Bayesian networks are quite easily translated to our new model.

Finally, recent work on efficient algorithms for Bayesian network computations using integer linear programming [11] seem to be readily applicable to our multi-networks. In particular, we have observed that the basic structure of the linear programming formulations are similar in many regards.

References

[1] Eugene Charniak and Eugene Santos, Jr. Dynamic map calculations for abduction. In *Proceedings of the AAAI Conference*, pages 552–557, 1992.

[2] Eugene Charniak and Solomon E. Shimony. Probabilistic semantics for cost based abduction. In *Proceedings of the AAAI Conference*, pages 106–111, 1990.

[3] Dan Geiger and David Heckerman. Advances in probabilistic reasoning. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 118–126, 1991.

[4] David Heckerman. *Probabilistic Similarity Networks*. The MIT Press, 1991.

[5] M. Henrion. Propagating uncertainty by logic sampling in bayes' networks. Technical report, Department of Engineering and Public Policy, Carnegie-Mellon University, 1986.

[6] Jerry R. Hobbs, Mark Stickel, Paul Martin, and Douglas Edwards. Interpretation as abduction. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, pages 95–103, 1988.

[7] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.

[8] Yun Peng and James A. Reggia. *Abductive Inference Models for Diagnostic Problem-Solving*. Springer-Verlag, 1990.

[9] Eugene Santos, Jr. On the generation of alternative explanations with implications for belief revision. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 337–347, 1991.

[10] Eugene Santos, Jr. *A Linear Constraint Satisfaction Approach for Abductive Reasoning*. PhD thesis, Department of Computer Science, Brown University, 1992.

[11] Eugene Santos, Jr. A fast hill-climbing approach without an energy function for finding mpe. In *Proceedings of the 5th IEEE International Conference on Tools with Artificial Intelligence*, 1993.

[12] Eugene Santos, Jr. A linear constraint satisfaction approach to cost-based abduction. To appear in Artificial Intelligence, 1993.

[13] Eugene Santos, Jr. Modeling cyclicity and generalized cost-based abduction using linear constraint satisfaction. To appear in Journal of Experimental and Theoretical Artificial Intelligence, 1994.

[14] Ross D. Shachter and Mark A. Peot. Simulation approaches to general probabilistic inference on belief networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 1989.

[15] Solomon E. Shimony. Algorithms for finding irrelevance-based map assignments to belief networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 1991.

[16] Solomon E. Shimony. Explanation, irrelevance and statistical independence. In *Proceedings of the AAAI Conference*, 1991.

[17] Solomon E. Shimony and Eugene Charniak. A new algorithm for finding map assignments to belief networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 1990.

[18] E. H. Shortliffe and B. G. Buchanan. A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23:351–379, 1975.

[19] Bon K. Sy. Reasoning mpe to multiply connected belief networks using message passing. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, 1992.

November 1993 Technical Report

COMPUTING WITH BAYESIAN MULTI-NETWORKS

Eugene Santos Jr.

Air Force Institute of Technology, WPAFB OH 45433-6583

AFIT/EN/TR93-10

Existing probabilistic approaches to automated reasoning impose severe restrictions on its knowledge representation scheme. Mainly, this is to ensure that there exists an effective inferencing algorithm. Unfortunately, this makes the application of these approaches to general domains quite difficult.
In this paper, we present a new model called Bayesian multi-networks which uses a rule-based organization of knowledge quite natural for human experts modeling various domains. Furthermore, strong probabilistic semantics help quantify the knowledge. Combined with the rich structure of rule-based approaches, a general inference engine for Bayesian multi-networks is developed.

probabilistic reasoning, constraint satisfaction, linear programming, temporal reasoning, abductive explanation 15

UNCLASSIFIED UNCLASSIFIED UNCLASSIFIED UL